Rethinking Intermediate Module Utilization in V2X End-to-End Autonomous Driving

Yiming Kan
Tongji University
Shanghai
yimingkan@tongji.edu.com

Huilin Yin*
Tongji University
Shanghai
yinhuilin@tongji.edu.cn

Daniel Watzenig
Graz University of Technology
Graz
daniel.watzenig@tugraz.at

Abstract

End-to-end autonomous driving has progressed rapidly, with vehicle-side models relying on perception or ego status. UniV2X has extended this paradigm to the Vehicle-to-Everything (V2X) domain, where the broader perceptual scope of V2X offers a more revealing context for revisiting the effective utilization of intermediate modules. Prior work has examined the utility of intermediate modules in vehicle-side models, with studies suggesting that historical trajectories or current ego status alone may suffice for achieving competitive performance on open-loop datasets. Our paper aims to revisit this assumption in the V2X setting. Using the UniV2X model as the baseline and the V2X-Seq dataset as the testbed, we examine the contribution of intermediate modules to the final planning output and explore the extent to which their utility is fully realized. Our study reveals that current end-to-end models tend to underutilize the guidance provided by intermediate modules to the planning stage, reflecting a lack of planning-oriented design. To address this issue, we propose ***Optimized Multi-Experts Guided Autonomous Driving (OMEGA)****, a functional integration mechanism that explicitly improves the contribution of intermediate modules to the planning process. Experimental results demonstrate that our approach significantly enhances the functional contribution of each intermediate component. Our findings suggest that performance limitations are not due to the lack of new modules but stem from the underutilization of existing ones, urging a reconsideration of current end-to-end design practices.*

1. Introduction

End-to-end autonomous driving has shown promising potential in jointly optimizing perception, prediction, and planning. However, whether intermediate modules are truly leveraged to inform downstream planning remains unclear.

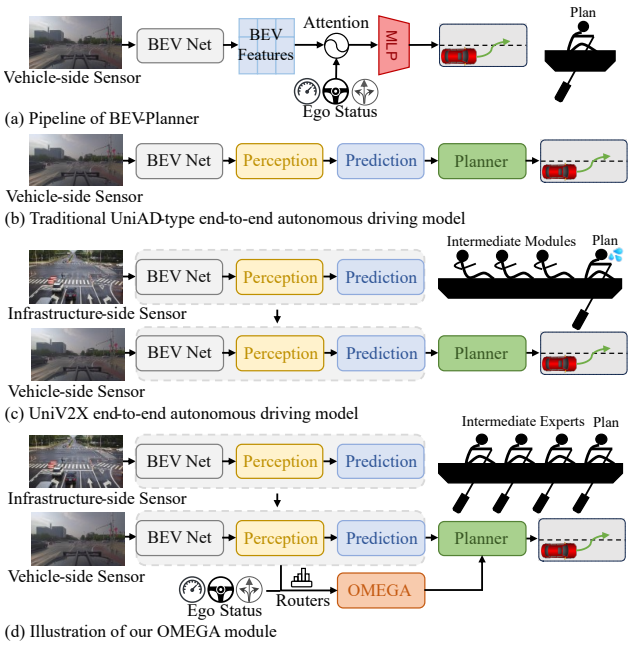


Figure 1. Comparison of architectures in end-to-end autonomous driving: (a) BEV-Planner relies only on BEV features and ego status. (b–c) UniAD and UniV2X stack multiple intermediate modules, which increases modeling capacity but often results in underutilized components. (d) OMEGA transforms intermediate modules into experts with routing guidance, enabling them to actively support planning.

Ablation studies in UniAD [4] show that stacking additional modules often degrades their effectiveness, suggesting substantial underutilization of intermediate representations. Some prior works have questioned the necessity of perception and prediction modules. Models such as BEV-Planner [7] and AD-MLP [15] achieve competitive planning performance using simplified designs (Figure 1(a)), but are highly sensitive to input variations, such as changes in ego speed [7]. In contrast, module-stacked architectures generally exhibit greater robustness. The key challenge, therefore, is to effectively combine the advantages of both ap-

*Corresponding author.

proaches. We believe this can be achieved by better exploiting these intermediate modules, ensuring they actively support rather than burden the planning module.

To further investigate the underutilization of intermediate modules, we focus on V2X cooperative scenarios, which offer two key advantages relevant to our study. First, V2X provides richer perceptual inputs, as infrastructure-side sensors complement vehicle-side blind spots. As shown in UniV2X [14], such cooperative design enhances perception, detection, and planning performance. This aligns with our objective, as these intermediate modules receive richer perceptual inputs, removing them is expected to have a greater impact on planning performance. This allows us to assess the true value of intermediate modules by observing the impact of masking their outputs. Second, the complexity of intermediate modules is exacerbated in UniV2X due to its dual-branch architecture spanning both vehicle and infrastructure sides. Inheriting the modular design of UniAD while extending it to a two-branch structure, UniV2X results in deeper and more burdensome pipelines (Figure 1(b,c)). We therefore adopt UniV2X as our baseline, as it exemplifies the challenges we aim to address.

In our experiments, however, we observe that although the V2X scenario provides richer perceptual information, the intermediate modules of UniV2X do not effectively contribute to the planning process. The planning module often relies on simplified representations, such as BEV features and ego status, bypassing deeper intermediate outputs. This phenomenon underscores the underutilization of existing modules, an overlooked bottleneck in current frameworks. To address this problem, we propose the **Optimized Multi-Experts Guided Autonomous Driving (OMEGA)** framework. Rather than introducing new modules or redesigning existing ones, OMEGA revitalizes intermediate modules by transforming them into dedicated experts. This is achieved through a Mixture-of-Experts (MoE) mechanism, which enables the planning module to actively select and attend to outputs from different intermediate modules. This framework aims to realize the potential of intermediate modules and alleviate the burden caused by increased architectural depth (see Figure 1 (d)). To further quantify the impact of intermediate modules, we also introduce the **Contribution Score**, a metric that measures how much each module contributes to the planning process.

Our key contributions are as follows:

- We propose a method to measure the contribution of intermediate modules to the planning process, which is often overlooked in existing frameworks.
- We present the OMEGA framework, which leverages Mixture-of-Experts (MoE) mechanism to activate the functional potential of intermediate modules.
- Our model improves the contribution scores of intermediate modules and reduces L2 Error (↓19.1%), Collision Rate (↓20.1%), and Off-Road Rate (↓47.3%) compared to UniV2X, while also lowering transmission cost.

2. Related Work

2.1. Vehicle-side End-to-End Autonomous Driving

Vehicle-side end-to-end autonomous driving aims to directly map raw on-board sensor data to control actions within a unified framework. Some models focus on modular integration via convolutional or transformer-based backbones, including UniAD [4], VAD [6], DriveTransformer [5], SparseDrive [10], MomAD [8], and BridgeAD [16]. A parallel line of research frames planning as a generative problem, employing approaches such as DiffusionDriveV2 [20], DiffusionPlanner [18], and GenAD [17] to enhance trajectory diversity. Meanwhile, Mixture-of-Experts (MoE) frameworks like DriveMoE [11] and ARTEMIS [2] introduce modular mechanisms for adapting to diverse driving scenarios. On the other hand, several studies explore minimal-input alternatives. AD-MLP [15] and BEV-Planner [7] demonstrate that MLPs using only ego-state inputs can achieve competitive open-loop performance, raising concerns about whether perception is fully utilized. These approaches reflect the increasing sophistication of vehicle-side frameworks, and our method follows the trend of adapting single-agent designs to cooperative V2X settings.

2.2. Cooperative End-to-End Autonomous Driving

V2X systems introduce cooperative perception by enabling information sharing between multiple agents, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, thereby offering a broader environmental context for autonomous driving. Early works such as Coopernaut [1] demonstrate that shared perception among networked vehicles can support end-to-end planning. UniV2X [14] extends the modular paradigm of UniAD to the V2X setting and serves as a benchmark on the V2X-Seq dataset [13]. V2XPnP [19] introduces a large-scale V2X dataset for multi-agent perception and prediction, enabling the study of spatio-temporal interactions in V2X scenarios. The MAP method proposed in [12] reveals that current end-to-end V2X models tend to underutilize the online mapping module. Recent work such as UniMM-V2X [9] explores MoE-enhanced multi-level fusion to selectively aggregate heterogeneous V2X features from perception to prediction.

While many V2X frameworks extend vehicle-side designs, the role of intermediate modules in cooperative planning remains insufficiently examined. We adopt UniV2X as a representative architecture to study whether such modules effectively contribute to planning. Although our ex-

periments are conducted in V2X settings, the insights apply broadly to both cooperative and single-agent autonomous driving models.

3. Method

3.1. Module Utilization Problem Formulation

In end-to-end autonomous driving models, perception and prediction modules are often structured as intermediate stages that pass high-dimensional features to the planning module. While these modules are architecturally connected to the planning module, their functional significance remains ambiguous. That is, the presence of feature flow does not guarantee that the planning decision actually depends on these intermediate outputs.

We define *module utilization* as the extent to which a module's output influences the final trajectory planning performance. A module is considered *functionally utilized* if its absence leads to measurable degradation in planning quality. This definition distinguishes between mere structural inclusion, in which a module is connected but not effectively used, and true functional contribution.

Formally, let the set of intermediate modules be $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$, where each M_i produces an output f_i. The planner P receives $\{f_i\}_{i=1}^n$ to generate the final trajectory $\hat{y} = P(f_1, \ldots, f_n)$. However, the degree to which each f_i influences $\hat{y}$ remains unclear.

Our goal is to move from structural inclusion to functional validation: we ask whether the final output $\hat{y}$ significantly changes when a particular f_i is removed. This leads to the diagnostic formulation introduced in the next subsection.

3.2. Module Contribution Metrics

To assess whether a module is functionally utilized, we propose a diagnostic approach based on performance degradation following module output suppression. Specifically, we zero out the output of each intermediate module before it is forwarded to the planning module and evaluate the resulting change in planning performance.

Each method is evaluated using three standard metrics: L2 Error, Collision Rate, and Off-road Rate. For each metric, we report the average over the 2.5 s, 3.5 s, and 4.5 s time horizons. To ensure comparability across metrics, we apply a normalized improvement metric to all three indicators. Following the evaluation protocol in [3], we set the reference values x_{ref} to 3.5 m (L2 error), 2% (collision rate), and 2.5% (off-road rate), and the corresponding improvement ranges x_{range} to 1.0 m, 1.5%, and 2.5%, respectively. The overall performance score is represented by:

$$P = \frac{x_{\text{ref}} - x}{x_{\text{range}}} \tag{1}$$

Let P_{base} denote the planning performance of the full model, and let P_{-M_i} be the performance when the output of module M_i is zeroed out. We define the *contribution score* C_i for module M_i as:

$$C_i = \frac{P_{\text{base}} - P_{-M_i}}{P_{\text{base}}} \times 100\% \tag{2}$$

Intuitively, a higher C_i implies greater functional reliance on module M_i, while a low or negative C_i suggests that the module's output has little to no effect on planning behavior. This formulation allows us to quantify the actual contribution of modules such as tracking, motion prediction, mapping and occupancy prediction modules in a unified manner.

In practice, we apply this probing strategy to the UniV2X model on the V2X-Seq dataset. During evaluation, we selectively zero out the output tensors of each intermediate module before their fusion or attention integration into the planning stream. By comparing the planning metrics across these ablations, we gain empirical insight into which components are actively used by the planner and which are structurally present but functionally dormant.

3.3. Architecture Overview

To address the underutilization of intermediate modules in current V2X end-to-end frameworks, we propose the **Optimized Multi-Experts Guided Autonomous Driving (OMEGA)** architecture. OMEGA converts the outputs of intermediate modules into domain-specific expert embeddings and introduces a structured routing and aggregation mechanism to guide planning more effectively.

Figure 2 presents an overview of our method. The architecture comprises three major components:

- **Plan Query Generator**, which augments the original planning query by incorporating current ego status information.
- **Expert Router**, a two-layer routing system that dynamically selects the most relevant experts based on current ego status and BEV context.
- **Experts Mixer**, a Transformer decoder that integrates expert outputs with the enhanced planning query Q^p.

Together, these components form a modular framework that enhances the functional impact of intermediate modules in trajectory planning.

3.4. Plan Query Generator

The Plan Query Generator augments the original planning query by explicitly incorporating the current ego status and driving command. The original query consists of a track query encoding past motion and a trajectory query encoding future intent. We encode the ego status through a linear layer and the command through an embedding layer, and concatenate them with the original query. (see Figure 3)

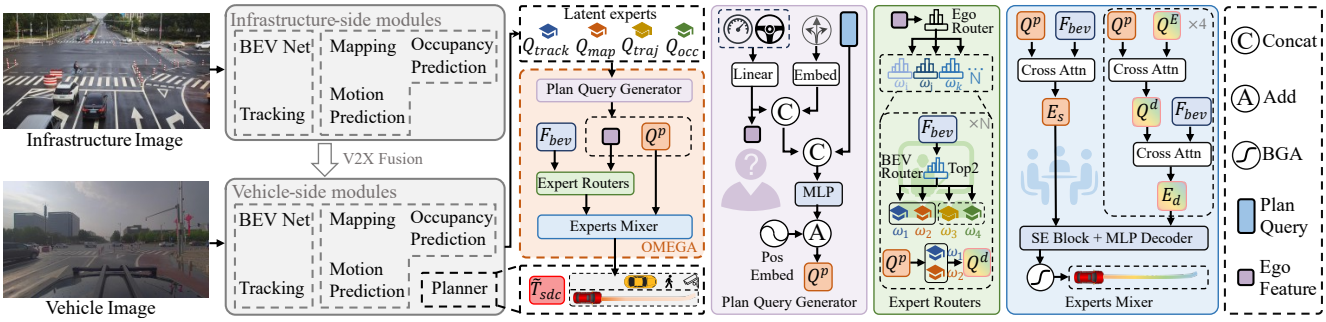


Figure 2. The overall architecture of the model with OMEGA. OMEGA is a mechanism that can be integrated into multi-module end-to-end autonomous driving models. It first extracts latent expert embeddings from intermediate module outputs and augments the original plan query with current ego status information to generate the enhanced plan query Q^p. The latent expert embeddings are then selectively routed by the Expert Routers module, based on ego features and BEV features. Finally, Q^p is refined through the Experts Mixer module to enable multi-expert collaborative planning. This approach addresses the critical challenge of underutilized intermediate modules in planning. In this figure, BGA refers to bivariate gaussian activation.

The fused representation is processed by an MLP followed by positional encoding to produce the enhanced planning query Q^p for downstream planning.

3.5. Experts Router

To effectively assign appropriate experts, we design a two-layer routing structure. The first layer, Ego Router, selects high-level driving strategies based on the current ego status, ensuring physically plausible decisions. The second layer, BEV Router, refines expert selection using environmental information. By separating ego-centric and scene-centric signals, this design avoids feature ambiguity and improves routing clarity.

3.5.1. Ego Router.

In our initial design, the ego router utilized discrete driving commands as input. However, visualization results revealed that this hard classification approach was unable to capture the gradual transitions between driving commands, particularly during the fade-in and fade-out phases. As a result, each command corresponded to a fixed trajectory pattern. As shown in Figure 4, when command-based weights are used, the ego router blindly follows the unchanged command, even if the vehicle is already transitioning to a new direction.

To overcome this limitation, we replace command inputs with a continuous representation of current ego status, including velocity, acceleration, yaw rate, and heading angle (encoded as $\cos(\theta)$ and $\sin(\theta)$). This allows the router to infer driving intent more fluidly, treating trajectory progression as a dynamic state rather than a fixed label. The improved ego router supports softer transitions and enhances trajectory realism.

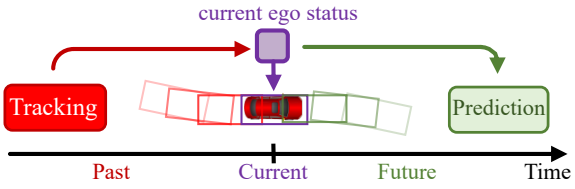


Figure 3. Connect the learned past trajectory and predicted future trajectory by injecting current ego status information

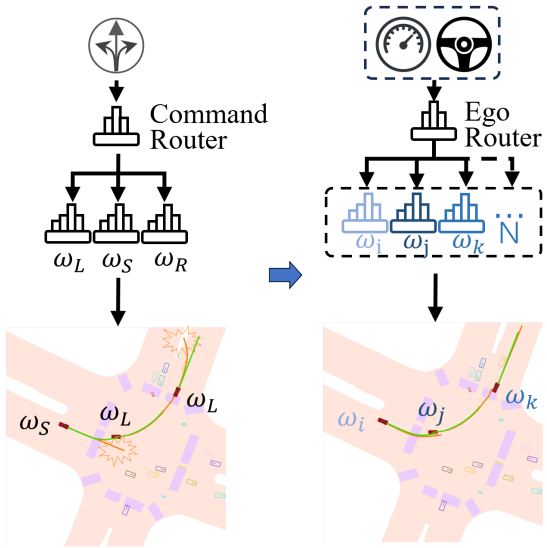


Figure 4. Comparison between command-based and ego-status-based routing. The ego router captures the smooth transition between turning and straight phases, avoiding rigid or overly curved trajectories.

Subsequently, the weight ω_s generated by the Ego Router is incorporated into the fusion process as follows:

$$E_d^{all} = \sum_{s \in \mathcal{I}} \omega_s \cdot \text{CrossAttn}(E_d^s, F_{\text{bev}}, F_{\text{bev}}) \qquad (3)$$

Here, $\mathcal{I} = \{i, j, k, \dots\}$ denotes the index set of all experts involved in the fusion, and its cardinality is set to $N = 5$.

3.5.2. BEV Router.

The BEV Router operates as a child module of the Ego Router. Conditioned on the ego status routing result, N BEV Routers are selected, and their outputs are aggregated through a weighted sum, where the weights are provided by the Ego Router. Although each BEV Router produces N_{experts} outputs, we adopt a sparse Mixture-of-Experts (MoE) mechanism to retain only the top-2 experts with the highest routing weights. These weights from BEV router are then used to compute a weighted combination of the corresponding latent domain-specific expert features Q_d.

The BEV Router produces four expert weights, from which we select the top two as follows:

$$\mathcal{T} = \underset{i \in \{1,2,3,4\}}{\arg\,\text{top-2}}\, \omega_i \tag{4}$$

Here, $\mathcal{T}$ denotes the set of top-2 indices from the expert routing weights $\omega_1, \omega_2, \omega_3, \omega_4$.

3.6. Experts Mixer

Our basic model comprises four experts: Q_{track} derived from tracking, Q_{traj} from motion prediction, Q_{seg} from mapping, and Q_{occ} from occupancy prediction. These are collectively referred to as Q_E, representing latent expert queries. The query used for decoding the trajectory is denoted as Q^p, which is generated by the Plan Query Generator.

$$Q_E^i \in \{Q_{\text{track}}, Q_{\text{map}}, Q_{\text{traj}}, Q_{\text{occ}}\}, \quad \text{for } i \in \mathcal{T}, \tag{5}$$

Each selected Q_E^i corresponds to one of the latent expert features from $Q_{\text{track}}, Q_{\text{map}}, Q_{\text{traj}}, Q_{\text{occ}}$.

3.6.1. Shared Expert.

To enhance training stability, we retain the BEV-Planner branch as a shared expert, serving as a stabilizing baseline within the Mixture-of-Experts (MoE) framework.

3.6.2. Domain-specific Expert.

Each domain-specific expert employs Q^p as the query, with Q_E serving as keys and values. Through a multi-head cross attention mechanism, the Transformer Decoder extracts the information from Q^p that aligns with the focus of each expert Q_E. Conceptually, this process "infuses" Q^p with the characteristics of Q_E, akin to tinting Q^p with the "color" of the expert. The resulting domain-aware query is denoted as Q_d. Subsequently, cross-attention with F_{BEV} yields the domain-specific output E_d, as shown in Equation 3.

$$Q_d = \sum_{i \in \mathcal{T}} \omega_i \cdot \text{CrossAttn}(Q^p, Q_E^i, Q_E^i), \tag{6}$$

The final domain-specific query Q_d is computed by applying cross-attention between the planning query Q^p and each selected expert, followed by a weighted sum using the corresponding ω_i.

3.6.3. Experts Mixture.

Drawing inspiration from the Squeeze-and-Excitation (SE) module in the computer vision domain, we incorporate an analogous gating mechanism into our MoE framework. Based on the routing coefficients produced by the Ego router and the BEV router, the top two outputs of domain-specific experts are first weighted and summed. Subsequently, the fused domain-specific expert and the shared expert, denoted as E_d and E_s, are passed to the SE module, which dynamically adjusts their relative contributions. This adaptive reweighting mechanism enables more effective and coherent integration of expert knowledge.

Let $E_d^s \in \mathbb{R}^{B\times 1\times C}$ denote the output of the s-th domain-specific expert with $s \in \mathcal{I}$, and let $E_s \in \mathbb{R}^{B\times 1\times C}$ denote the output of the shared expert. To obtain scalar descriptors for each expert, we compute the average over the batch and channel dimensions:

$$\hat{e}_d^s = \frac{1}{B \cdot C} \sum_{b=1}^{B} \sum_{c=1}^{C} (E_d^s)_{(b,1,c)} \tag{7}$$

$$\hat{e}_s = \frac{1}{B \cdot C} \sum_{b=1}^{B} \sum_{c=1}^{C} (E_s)_{(b,1,c)} \tag{8}$$

These scalar descriptors, consisting of $\{\hat{e}_d^s\}_{s\in\mathcal{I}}$ from domain-specific experts and $\hat{e}_s$ from the shared expert, are concatenated into a vector:

$$\mathbf{s} = [\hat{e}_i, \hat{e}_j, \hat{e}_k, \ldots, \hat{e}_s]^\top \in \mathbb{R}^{N_{\text{experts}}+1} \tag{9}$$

where $N_{\text{experts}} = 4$ corresponding to tracking, motion prediction, mapping, and occupancy prediction.

This vector is passed through a two-layer fully connected gating network:

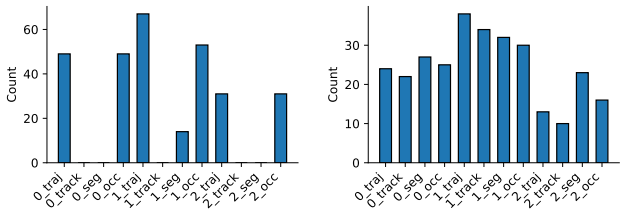
$$\mathbf{a} = \text{ReLU}(\mathbf{W_1 s}) \tag{10}$$

$$\mathbf{w} = \text{Softmax}(\mathbf{W_2 a}) \tag{11}$$

where $w = [w_i, w_j, w_k, \ldots, w_s]^\top \in \mathbb{R}^{N_{\text{experts}}+1}$ is the learned weight vector over all experts, and W_1 and W_2 are learnable weight matrices.

The final fused representation is then computed as:

$$\tilde{E} = w_s \cdot E_s + \sum_{s\in\mathcal{I}} w_d^s \cdot E_d^s \tag{12}$$



(a) w/o temperature (b) w temperature

Figure 5. (a) and (b) show the expert selection frequency comparison without and with temperature when $N = 3$.

To ensure that all experts receive sufficient training during the early stages, we apply a temperature-controlled softmax function to smooth the assignment weights, allowing each expert's corresponding transformer layer to be effectively trained, as illustrated in Figure 5.

4. Experiments

4.1. Experimental Settings

Datasets and Metrics. The V2X-Seq dataset is a large-scale, real-world benchmark designed for cooperative autonomous driving. It comprises 72,890 frames of synchronized 2D images and 3D LiDAR point clouds with ground-truth annotations, captured at a frequency of 2 Hz. We conduct our experiments on the V2X-Seq-SPD subset, which includes over 15,000 sequential frames across 95 distinct scenes and serves as an official benchmark for evaluating end-to-end autonomous driving models in the V2X context. Planning performance is assessed using three primary metrics: L2 Error, Collision Rate, and Off-Road Rate. In addition, transmission cost is reported to evaluate communication efficiency.

Training Details. The model predicts the future trajectory of the ego vehicle over a 5-second horizon, corresponding to 10 discrete timesteps. All experiments are conducted on two NVIDIA RTX A800 GPUs. We initialize the model using the publicly released `univ2x_coop_e2e_stg1` checkpoint from UniV2X and adopt the same set of hyperparameters to ensure comparability. The total training time is approximately 50 hours.

4.2. Module Contribution Studies

To validate our hypothesis about the underutilization of intermediate modules in existing frameworks, we conducted a two-stage ablation study. First, we independently masked each intermediate module in the original UniV2X model and observed only marginal degradation in planning performance, suggesting their limited contribution under the original design. Second, after introducing our proposed framework OMEGA, we repeated the masking process. This time, the removal of intermediate modules led to a significant drop in performance, demonstrating that when fully utilized, each module plays a critical role in the final trajectory planning.

We first compute the performance scores of each test model across three evaluation metrics: L2 error, collision rate, and Off-Road Rate. The contribution score for each metric is then calculated as the percentage difference between the performance score of the ablated model and that of the full model. The total contribution score for each ablation experiment is obtained by summing the contribution scores from the three metrics. In Table 1, contribution scores are rounded to three significant digits, while other values are reported with two decimal places.

4.2.1. Contribution Studies on the Baseline Model.

We adopt the normalized performance and contribution score formulations from Section 3.2. Experiments on the baseline model reveal that zeroing out certain intermediate modules leads to comparable or even improved Avg. L2 Error compared to the full model, suggesting that these modules contribute little, and in some cases may even impair trajectory accuracy. For Collision Rate, the Occ Module shows the highest contribution score, which is expected since its output is used for post-processing outside the core model. In contrast, masking the Seg Module yields negligible change in Off-Road Rate, indicating that map information from the segmentation network is not effectively utilized by the planner.

Overall, the Seg Module is the only one with a positive total contribution score when aggregated across all metrics. These findings highlight a substantial underutilization of intermediate modules in the baseline design.

4.2.2. Contribution Studies on Our Model.

With the proposed OMEGA framework, we observe that the Motion Module exhibits the highest contribution to the L2 metric. This aligns with domain expectations, in contrast to the baseline model where removing the Motion Module yields minimal impact. Similarly, the Occ Module contributes most significantly to the Collision Rate. Notably, we remove the external post-processing step from the baseline, enabling the Occ Module to participate directly in trajectory planning as a latent expert within the model. Regarding the Off-Road Rate, the Seg Module demonstrates a substantial contribution. This corrects the counterintuitive behavior in the baseline model, where masking the output of Seg Module had no effect on off-road performance. The result indicates that our improved model is capable of effectively leveraging the information produced by intermediate modules.

Overall, the Total Contribution Score across all modules is positive, suggesting that OMEGA successfully addresses the issue of underutilized intermediate modules by promoting their functional integration into the planning process.

Moreover, the ablation results presented in Table 3 clearly demonstrate the effectiveness and necessity of each component in the proposed OMEGA framework, confirming that every module contributes meaningfully to the cooperative planning performance.

4.2.3. Comparison with Other Methods.

We compared our model with several existing baselines in Table 2 and highlight two representative types: a simple end-to-end autonomous driving model without intermediate modules, and a UniAD-like model with extensive inter-

Ablation Setting	L2 Error (m) ↓	Col. Rate (%) ↓	Off-Road Rate (%) ↓	Perf. ↑				Contr. (%) ↑			
	Avg.	Avg.	Avg.	L2	Col	Off	Total	L2	Col	Off	Total
Full UniV2X*	3.46	0.34	0.74	0.04	1.11	0.70	1.85	—	—	—	—
- Track	3.43	0.74	0.89	0.07	0.84	0.64	1.55	-75.0	24.3	8.57	-42.2
- Seg	3.46	0.44	0.74	0.04	1.04	0.70	1.78	0.00	6.31	0.00	6.31
- Motion	3.43	0.74	0.94	0.07	0.84	0.62	1.53	-75.0	24.3	11.4	-39.3
- Occ †	3.43	1.08	0.89	0.07	0.61	0.64	1.32	-75.0	45.0	8.57	-21.4
Full OMEGA	**2.80**	**0.27**	**0.39**	**0.70**	**1.15**	**0.84**	**2.69**	—	—	—	—
- Track	2.91	0.34	0.86	0.59	1.11	0.64	2.34	15.7	3.48	23.8	43.0
- Seg	3.18	0.30	1.59	0.32	1.13	0.36	1.81	54.3	1.74	**57.1**	**113**
- Motion	3.32	0.40	0.66	0.18	1.07	0.74	1.99	**74.3**	6.96	11.9	93.2
- Occ	2.92	0.56	0.63	0.58	0.96	0.75	2.29	17.1	**16.5**	10.7	44.3

Table 1. Comparison of ablation experiments for UniV2X and OMEGA models. We zeroed out the queries of the different intermediate modules and retrained the collaborative planning stage. * indicates re-training was done on our own hardware. † indicates occupancy output was only used in post-process. Perf. = Performance Score, Contr. = Contribution Score.

Method	L2 Error (m) ↓				Col. Rate (%) ↓				Off-Road Rate (%) ↓				Transm. Cost (BPS) ↓
	2.5s	3.5s	4.5s	Avg.	2.5s	3.5s	4.5s	Avg.	2.5s	3.5s	4.5s	Avg.	
CooperNaut [1]	3.84	5.33	6.87	5.35	0.44	1.33	1.93	1.23	**0.15**	**0.15**	1.33	0.54	8.19×10^7
UniV2X No Fusion	2.58	3.37	4.36	3.44	0.15	1.04	1.48	1.08	0.44	0.56	2.22	1.08	0
UniV2X Vanilla	2.33	3.69	5.12	3.71	0.59	2.07	3.70	2.12	0.15	1.33	4.74	2.07	8.19×10^7
UniV2X BEV Fusion	2.31	3.29	4.31	3.30	**0.00**	1.04	1.48	0.83	0.44	0.44	1.91	0.93	8.19×10^7
UniV2X†	2.55	3.35	4.47	3.46	**0.00**	0.44	**0.59**	0.34	0.30	0.74	1.19	0.74	8.09×10^5
UniV2X + Ego Status†	2.41	3.19	4.24	3.28	0.74	1.04	0.74	0.84	1.33	1.33	1.18	1.28	8.09×10^5
BEV-Planner [7]	2.29	3.12	4.11	3.17	**0.00**	0.74	1.04	0.59	0.44	0.74	1.11	0.94	8.19×10^7
OMEGA (Ours)	**1.88**	**2.78**	**3.74**	**2.80**	**0.00**	**0.15**	0.67	**0.27**	**0.15**	**0.15**	**0.86**	**0.39**	$\mathbf{1.69 \times 10^5}$*

Table 2. Comparison of L2 error, collision rate, Off-Road Rate, and transmission cost across different methods. *For OMEGA, we do not apply the same post-processing as in UniV2X; thus, the transmission cost of occupancy probability maps is omitted. All other UniV2X data, except for the value marked with †, are cited from [14].

Settings	L2(m) ↓	Col.(%) ↓	Off.(%) ↓	Perf. ↑
wo EM	3.51	0.59	0.89	1.57
wo ER	3.17	0.37	0.67	2.15
wo PQG	2.96	0.31	0.61	2.43
wo temperature	2.87	0.29	0.46	2.59
Full model	**2.80**	**0.27**	**0.39**	**2.69**

Table 3. Average value, PQG stands for Plan Query Generator, ER stands for Experts Router, EM stands for Experts Mixer.

mediate processing. Our model outperforms both, showing that making full use of intermediate modules is more effective than simply removing them. This addresses the concern raised by [7] and [15], which argue that an accumulation of intermediate modules may burden planning. To ensure a fair comparison and isolate the effect of our design, we added the same ego status information to the baseline model, demonstrating that the performance gain stems from our architecture rather than additional input features.

4.2.4. Qualitative Analysis.

Figure 6 presents a representative example at a complex intersection. The green trajectory corresponds to the ground-truth, while the orange trajectory shows the output of our model. The close alignment between them indicates that the model generates accurate and stable trajectories in complex urban scenarios.

In this example, the vehicle smoothly navigates through the transitional phases of entering and exiting the intersection, aligning with the design objective described in Section 3.5. This suggests that the model handles continuous driving states, rather than reacting only to discrete command switches. In the End Turning phase of the TURN RIGHT case, the predicted trajectory avoids a nearby vehicle near the end of the planning horizon, while the ground-truth does not. This indicates that the model produces safer

Figure 6. Qualitative comparison between the model trajectory (orange) and the ground-truth trajectory (green) in a complex turning scenario.

maneuvers by accounting for potential risks, rather than strictly imitating demonstration trajectories. Such behavior reflects improved decision adaptability in complex traffic situations.

5. Conclusion

In this work, we propose the **Optimized Multi-Experts Guided Autonomous Driving (OMEGA)** framework, which builds upon the UniV2X baseline by emphasizing effective utilization of each intermediate module. We revisit the role of intermediate modules and propose a framework that effectively treats each of them as an expert, and selects experts who can optimize the current planning decision based on current ego status and BEV features. Through the MoE concept, each module can dynamically maximize its domain-specific contribution. Consequently, without relying on any post-processing steps, our model achieves competitive results on the V2X-Seq dataset. These findings highlight the importance of fully utilizing intermediate modules in end-to-end autonomous driving models. We believe that our rethinking of the V2X end-to-end autonomous driving paradigm offers valuable insights into the design and improvement of future autonomous driving systems.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62433014 and No. 62133011.

References

[1] Junning Cui, Han Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17252–17262, 2022. 2, 7

[2] Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. ARTEMIS: Autoregressive End-to-End Trajectory Planning with Mixture of Experts for Autonomous Driving. *IEEE Robotics and Automation Letters*, pages 226–233, 2026. 2

[3] Ruiyang Hao, Haibao Yu, Jiaru Zhong, Chuanye Wang, Jiahao Wang, Yiming Kan, Wenxian Yang, Siqi Fan, Huilin Yin, Jianing Qiu, Yao Mu, Jiankai Sun, Li Chen, Walter Zimmer, Dandan Zhang, Shanghang Zhang, Mac Schwager, Wei Huang, Xiaobo Zhang, Ping Luo, and Zaiqing Nie. Research Challenges and Progress in the End-to-End V2X Cooperative Autonomous Driving Competition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1849–1860, 2025. 3

[4] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-Oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17853–17862, 2023. 1, 2

[5] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: Unified Transformer for Scalable End-to-End Autonomous Driving. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 2

[6] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–10, 2023. 2

[7] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14864–14873, 2024. 1, 2, 7

[8] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[9] Ziyi Song, Chen Xia, Chenbing Wang, Haibao Yu,

Sheng Zhou, and Zhisheng Niu. UniMM-V2X: MoE-Enhanced Multi-Level Fusion for End-to-End Cooperative Autonomous Driving. *arXiv preprint arXiv:2511.09013*, 2025. 2

[10] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 2

[11] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. DriveMoE: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025. 2

[12] Huilin Yin, Yiming Kan, and Daniel Watzenig. MAP: End-to-End Autonomous Driving with Map-Assisted Planning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1823–1830, 2025. 2

[13] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, Juan Song, Jirui Yuan, Ping Luo, and Zaiqing Nie. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5486–5495, 2023. 2

[14] Haibao Yu, Wenxian Yang, Jiaru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. End-to-end autonomous driving through v2x cooperation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9598–9606, 2024. 2, 7

[15] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023. 1, 2, 7

[16] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6854–6863, 2025. 2

[17] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *Computer Vision – ECCV 2024*, pages 94–111, Cham, 2025. Springer. 2

[18] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-Based Planning for Autonomous Driving with Flexible Guidance. In *International Conference on Learning Representations (ICLR)*, 2025. https://openreview.net/forum?id=wM2sfVgMDH. 2

[19] Zewei Zhou, Hao Xiang, Zhaoliang Zheng, Seth Z. Zhao, Mingyue Lei, Yun Zhang, Tianhui Cai, Xinyi Liu, Johnson Liu, Maheswari Bajji, Xin Xia, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. V2XPnP: Vehicle-to-Everything Spatio-Temporal Fusion for Multi-Agent Perception and Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4501–4511, 2025. 2

[20] Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. DiffusionDriveV2: Reinforcement Learning-Constrained Truncated Diffusion Modeling in End-to-End Autonomous Driving. *arXiv preprint arXiv:2512.07745*, 2025. 2